



JACKSON LAWRENCE  
00000070612

LOUIS GABRIEL HERNANDES  
00000070250

# ANT COLONY OPTIMIZATION



OSCAR JIRO HARLISON  
00000072786

IGNATIUS STEVEN  
00000070642



## DEFINITION

- **Ant Colony Optimization (ACO)** adalah sebuah algoritma metaheuristik yang menggunakan teknik **Swarm Intelligence (SI)** untuk memecahkan masalah-masalah optimasi
- Umumnya, masalah optimasi yang ingin diselesaikan dengan ACO perlu berupa **masalah optimasi diskrit** dan **direduksi menjadi bentuk graph** terlebih dahulu, di mana solusinya berupa **rute terbaik graph**
- Dikenalkan oleh Marco Dorigo pada 1992, ACO terinspirasi dari **perilaku koloni semut** dalam mencari sumber makanan

## pheromones

- Semut berkomunikasi dengan sesama dengan zat kimia yang disebut **feromon** yang akan dikeluarkan dalam bentuk jejak di tanah yang dapat diikuti dan diterima semut lain
- Komunikasi melalui jejak feromon membantu koloni semut perlahan menemukan **rute terdekat dari sarang ke sumber makanan**



## WHY USE ACO?

- Cocok untuk menyelesaikan masalah **optimasi diskrit** seperti kombinatorial, i.e. TSP, VRP
- Berupa algoritma terdistribusi dengan banyak agen sehingga **dapat diparalelisasi** dan **mengurangi kesempatan tersangkut di optimum lokal**
- **Fleksibel** dengan bertambahnya jumlah variabel atau batasan seperti masalah alokasi sumber daya dan penjadwalan

## APPLICATIONS

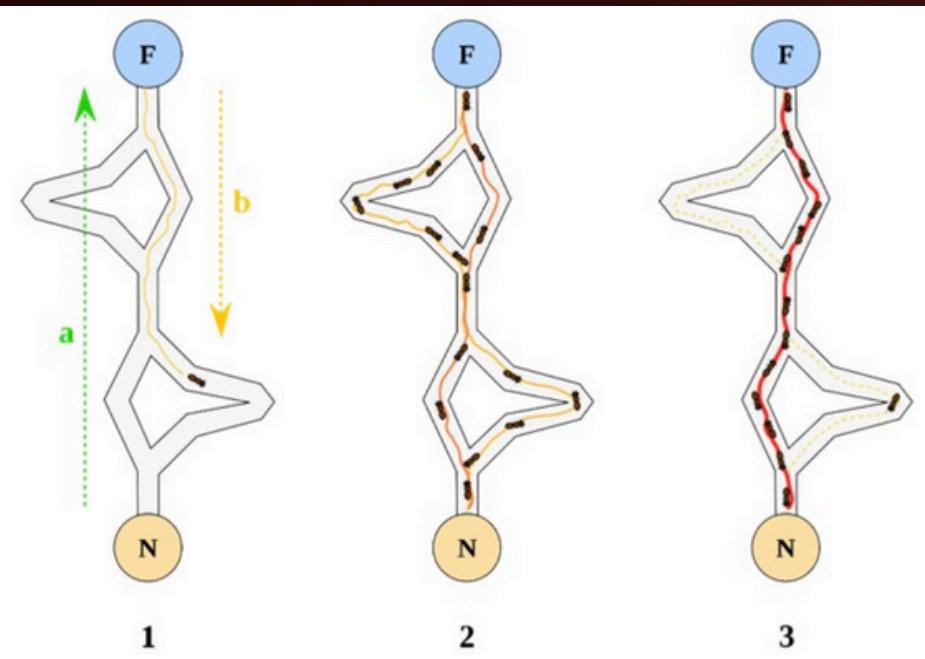
- Traveling salesman
- Vehicle routing
- Scheduling problems
- Network routing
- Constraint satisfaction
- Telecommunication networks
- Robotic path planning
- Protein folding
- Supply chain optimization
- Image processing

## HOWEVER...

- **Konvergensi yang lama:** ACO membutuhkan banyak iterasi untuk menemukan solusi mendekati optimal, sehingga bisa lambat untuk masalah besar
- **Parameter sensitif:** Hasil ACO sangat bergantung pada penentuan parameter, seperti tingkat evaporasi feromon dan jumlah semut
- **Tidak cocok untuk masalah kontinu:** ACO murni terbatas dalam pemecahan optimasi kontinu karena didesain untuk diskrit, kecuali dihibridasi dengan metode lain
- **Komputasi intensif:** ACO dapat membutuhkan memori dan daya komputasi yang besar, terutama untuk masalah dengan banyak variabel.



## HOW IT WORKS



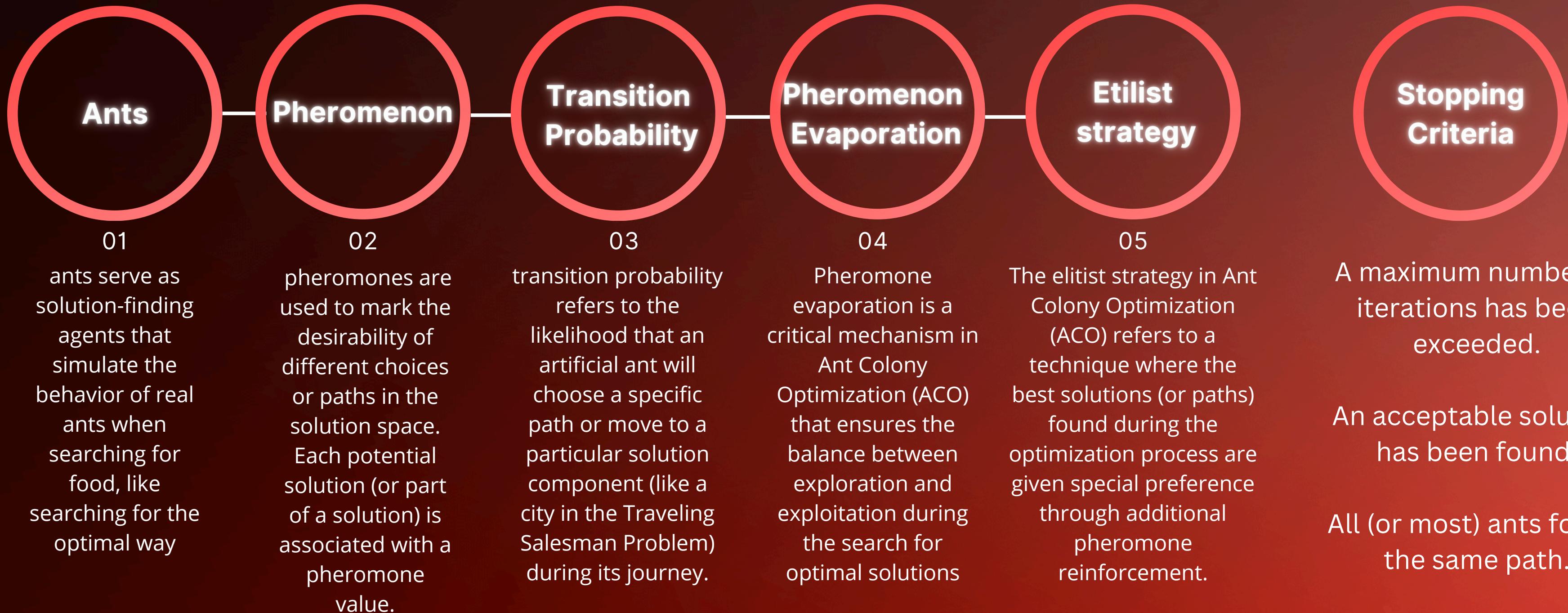
- Awalnya, semut akan mencari makanan secara acak, sehingga membuka banyak rute yang berbeda
- Saat semut sudah menemukan dan membawa balik makanan, mereka akan meninggalkan jejak feromon dari sumber yang dapat memandu semut-semut lain
- Karena feromon dapat berevaporasi, rute terpendek dengan deposit feromon terbanyak akan bertahan paling lama dan diprioritaskan semut lain

## OVERALL CONCEPT

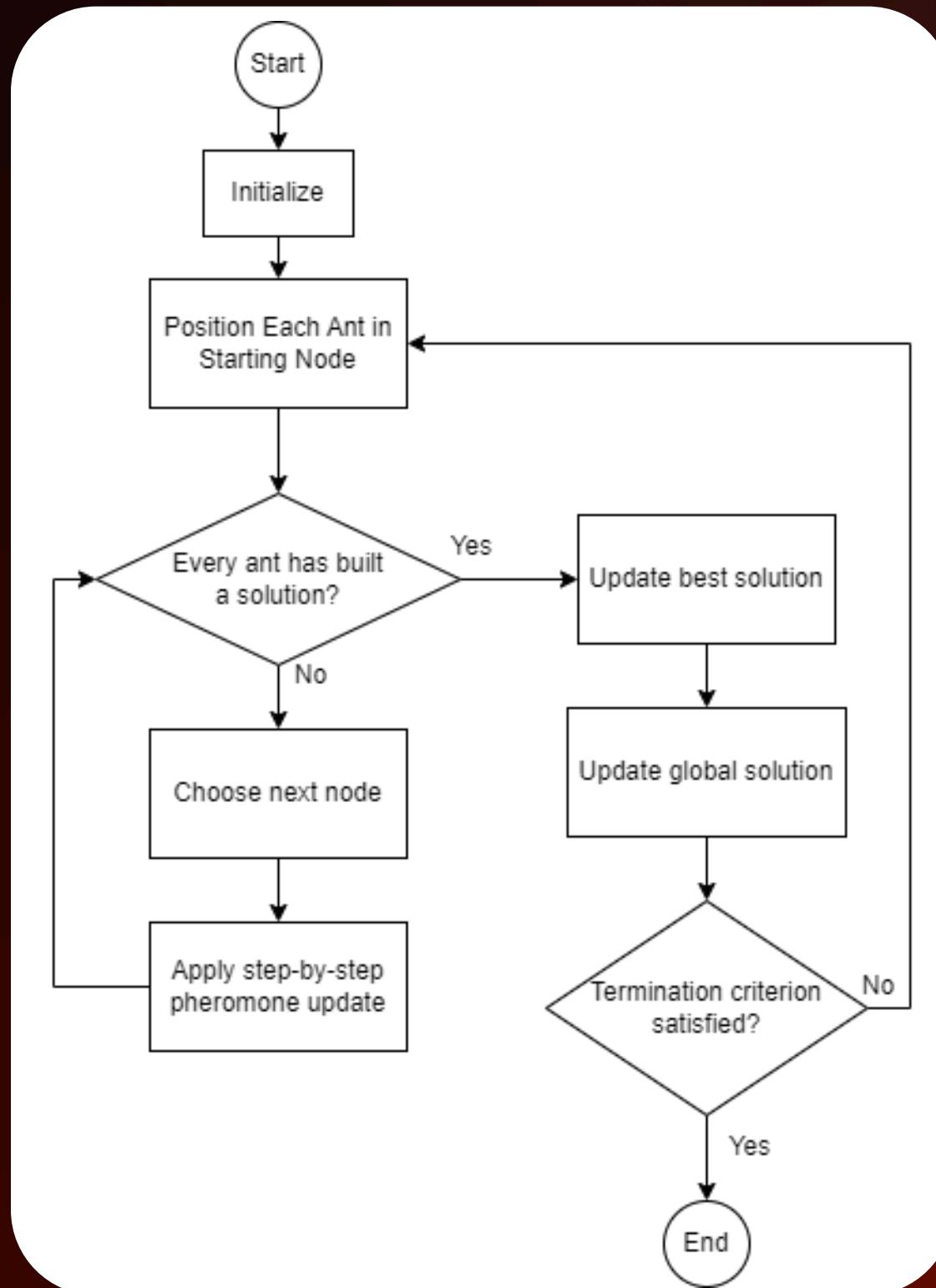




# KEY COMPONENTS & STOPPING CRITERIA



# PSEUDOCODE



## Ant Colony Optimization Algorithm

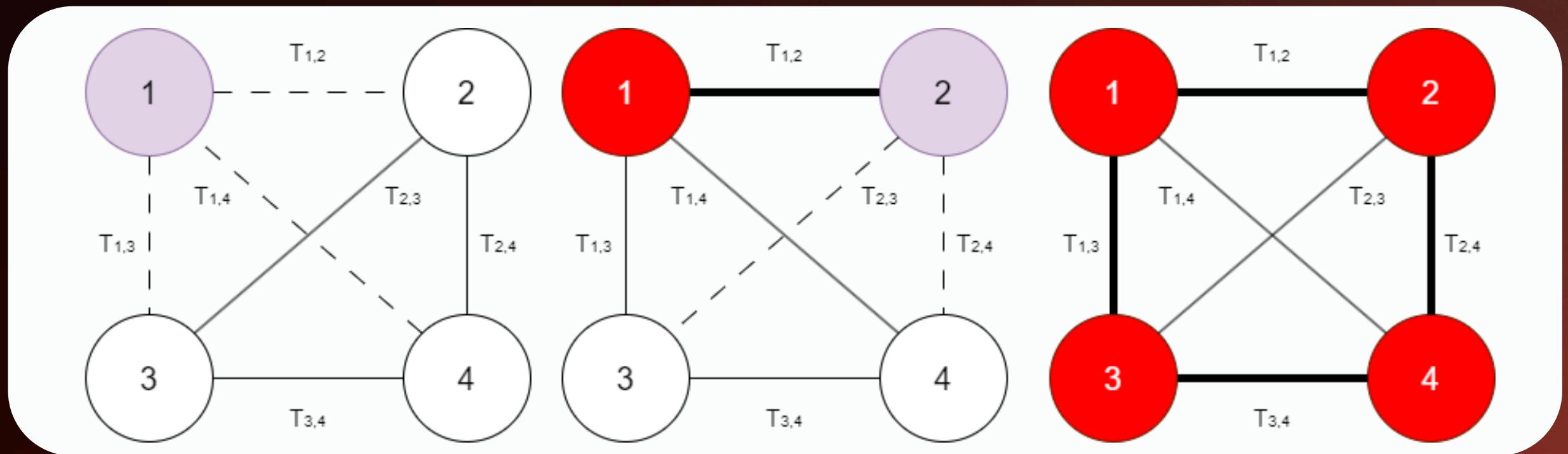
```

Initialise  $\tau_{ij}(0)$  to small random values; Let  $t = 0$ ;
Place  $n_k$  ants on the origin node;
while STOP-CRIT do
  for each ant  $k = 1, \dots, n_k$  do
     $x^k(t) = \emptyset$ ;
    While destination has not been reached do
      Select next node based on translation probability  $p_{ij}^k(t)$ ;
      Add  $(i,j)$  to path  $x^k(t)$ ;
    end
    Remove all loops from  $x^k(t)$ ;
    Calculate  $f(x^k(t))$ ;
  end
  for each edge  $(i,j)$  of the graph do
    Reduce the pheromone,  $\tau_{ij}(t) \leftarrow (1 - \rho)\tau_{ij}(t)$ ;
  end
  for each edge  $(i,j)$  of the graph do
    Update  $\tau_{ij}(t)$ , i.e.,  $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t)$ ;
  end
   $t \leftarrow t + 1$ ;
end
  
```

<https://github.com/KriezAlf/ExSys/blob/main/ACOA>



# HOW ALGORITHM WORKS



**Node 1:**  $N_1^k(t) = \{2, 3, 4\}$

$$p_{12}^k(t) = \frac{\tau_{12}^\alpha(t)}{\tau_{12}^\alpha(t) + \tau_{13}^\alpha(t) + \tau_{14}^\alpha(t)}$$

$$p_{13}^k(t) = \frac{\tau_{13}^\alpha(t)}{\tau_{12}^\alpha(t) + \tau_{13}^\alpha(t) + \tau_{14}^\alpha(t)}$$

$$p_{14}^k(t) = \frac{\tau_{14}^\alpha(t)}{\tau_{12}^\alpha(t) + \tau_{13}^\alpha(t) + \tau_{14}^\alpha(t)}$$

**Node 2:**  $N_2^k(t) = \{3, 4\}$

$$p_{23}^k(t) = \frac{\tau_{23}^\alpha(t)}{\tau_{23}^\alpha(t) + \tau_{24}^\alpha(t)}$$

$$p_{24}^k(t) = \frac{\tau_{24}^\alpha(t)}{\tau_{23}^\alpha(t) + \tau_{24}^\alpha(t)}$$

**Node 4:**  $N_4^k(t) = \{3\}$

$$p_{43}^k(t) = \frac{\tau_{43}^\alpha(t)}{\tau_{43}^\alpha(t)} = 1$$

TIME  
COMPLEXITY

$O(I * A * N^2)$

SPACE  
COMPLEXITY

$O(N^2 + A * N)$

## Informations:

I : Number of iterations

A : Number of ants

N : Number of nodes or cities



## TRANSITION PROBABILITY

Method 1:

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \times \eta_{ij}^\beta(t)}{\sum_{u \in N_i^k(t)} \tau_{iu}^\alpha(t) \times \eta_{iu}^\beta(t)}, & \text{if } j \in N_i^k(t) \\ 0, & \text{otherwise} \end{cases}$$

Method 2:

$$p_{ij}^k(t) = \begin{cases} \frac{\alpha \times \tau_{ij}(t) + (1 - \alpha) \times \eta_{ij}(t)}{\sum_{u \in N_i^k(t)} \alpha \times \tau_{iu}(t) + (1 - \alpha) \times \eta_{iu}(t)}, & \text{if } j \in N_i^k(t) \\ 0, & \text{otherwise} \end{cases}$$

## PHEROMONE EVAPORATION

$$\tau_{ij}(t) \leftarrow (1 - \rho) \times \tau_{ij}(t)$$

$$\eta_{ij}(t) = \frac{1}{d_{ij}(t)}$$

## UPDATE OF PHEROMONE INTENSITY / CONCENTRATION

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta \tau_{ij}^k(t)$$

$$\tau_{ij}(t+1) = (1 - \rho) \times \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta \tau_{ij}^k(t)$$

atau

### Informations:

- (i, j) : Edge / Garis dari node i ke node j
- t : Iterasi ke-t
- $d_{ij}(t)$  : Jarak antara node i dan j pada iterasi t
- $\eta_{ij}(t)$  : Nilai Heuristic antara node i dan j pada iterasi t
- $\tau_{ij}(t)$  : Konsentrasi Pheromone antara node i dan j pada iterasi t
- $n_k$  : Number of ants
- $\alpha > 0$  : Mengontrol pengaruh dari pheromone
- $\beta > 0$  : Mengontrol pengaruh dari nilai heuristic
- $\rho$  : Evaporation rate ( $0 < \rho \leq 1$ )



## ANT-CYCLE AS / PHEROMONON DIFFERENCES

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{f(x^k(t))}, & \text{if edge}(i, j) \text{ occurs in path } x^k(t) \\ 0, & \text{otherwise} \end{cases}$$

## ANT-DENSITY AS

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q, & \text{if edge}(i, j) \text{ occurs in path } x^k(t) \\ 0, & \text{otherwise} \end{cases}$$

## ANT-QUANTITY AS

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{d_{ij}(t)}, & \text{if edge}(i, j) \text{ occurs in path } x^k(t) \\ 0, & \text{otherwise} \end{cases}$$

$$d_{ij}(t) = d_{ji}(t)$$

$$f(\bar{x}(t)) = \min_{k=1, \dots, n_k} f(\bar{x}^k(t))$$

$$\tau_{ij}(t) = \tau_{ji}(t)$$

## ETILIST STRATEGY AS

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta\tau_{ij}^k(t) + n_e \times \Delta\tau_{ij}^e(t)$$

$$\Delta\tau_{ij}^e(t) = \begin{cases} \frac{Q}{f(\bar{x}(t))}, & \text{if edge}(i, j) \text{ occurs in path } \bar{x}(t) \\ 0, & \text{otherwise} \end{cases}$$

## ANT COLONY SYSTEM

$$j = \begin{cases} \arg \max_{u \in N_i^k(t)} \{\tau_{iu}(t) \times \eta_{iu}^\beta(t)\}, & \text{if } r \leq r_0 \\ J, & \text{if } r > r_0 \end{cases}$$

$$\tau_{ij}(t) \leftarrow (1 - \rho_L) \times \tau_{ij}(t) + \rho_L \times \tau_0$$

$$\tau_{ij}(t+1) = (1 - \rho_G) \tau_{ij}(t) + \rho_G \times \Delta\tau_{ij}(t)$$

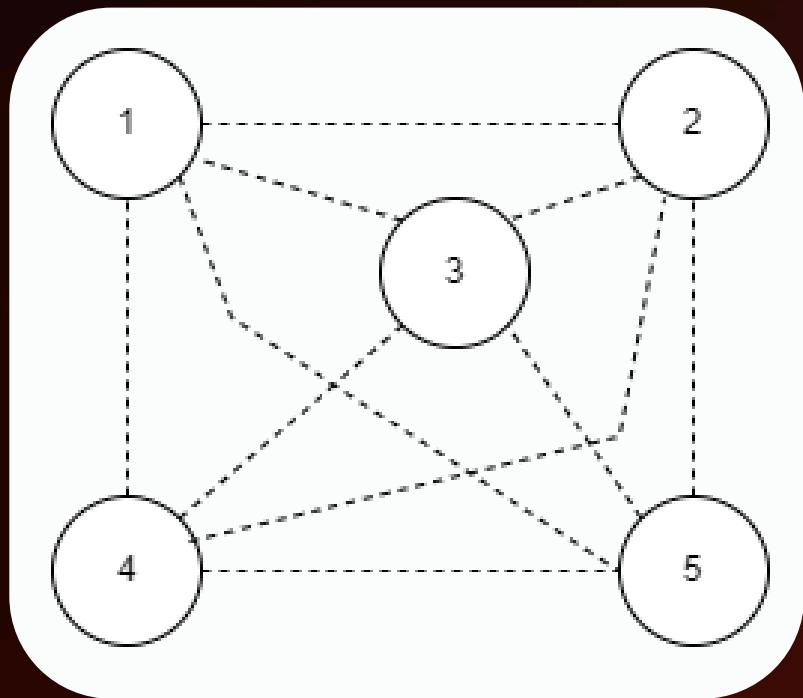
### Informations:

- $x^k(t)$  : Solusi dari semut k
- $\bar{x}(t)$  : Current best path / solution di iterasi t
- $r > r_0$  : Algorithm explores

- $f(x^k(t))$  : Kualitas dari solusi / jumlah jarak
- $n_e$  : Banyak elite ants
- $Q > 0$  : Constant
- $r \leq r_0$  : Algorithm exploits



# TRAVELING SALESMAN PROBLEM



Misalkan tiap semut ditempatkan pada tiap node dan ingin menjalankan tiap rute

Nilai Jarak  $d_{ij}(t)$

$i \setminus j$	1	2	3	4	5
1	0	39.1	80.6	36.9	35.1
2	39.1	0	101	27	24.9
3	80.6	101	0	84.9	85.2
4	36.9	27	84.9	0	4
5	35.1	24.9	85.2	4	0

Misalkan beberapa nilai awal:

- $\tau_{ij}(t=0) = 0.021$
- $\alpha = 1$
- $\beta = 1$
- $\rho = 0.9$

$$p_{ij}^k(t) = \frac{\tau_{ij}(t) * \eta_{ij}(t)}{\sum \tau_{iu}(t) * \eta_{iu}(t)}$$

$$p_{12}^1(0) = \frac{\tau_{12}(0) * \eta_{12}(0)}{\tau_{11}(0)*\eta_{11}(0) + \tau_{12}(0)*\eta_{12}(0) + \tau_{13}(0)*\eta_{13}(0) + \tau_{14}(0)*\eta_{14}(0) + \tau_{15}(0)*\eta_{15}(0)}$$

$$p_{12}^1(0) = \frac{0.021 * 0.026}{0*0.021 + 0.026*0.021 + 0.012*0.021 + 0.027*0.021 + 0.028*0.021}$$

$$= 0.28$$

Nilai Visibilitas  $\eta_{ij}(t) = 1 / d_{ij}(t)$

$i \setminus j$	1	2	3	4	5
1	0	0.026	0.012	0.027	0.028
2	0.026	0	0.01	0.037	0.111
3	0.012	0.01	0	0.012	0.012
4	0.027	0.037	0.012	0	0.25
5	0.028	0.04	0.012	0.25	0

Hasil pergerakan 5 semut dari node awal ke rute selanjutnya

Node Awal	Probabilitas					Node Tujuan	Memori
	1	2	3	4	5		
1	0	0.28	0.129	0.29	<b>0.301</b>	5	$1 \rightarrow 5$
2	0.23	0	0.088	0.327	<b>0.354</b>	5	$2 \rightarrow 5$
3	<b>0.261</b>	0.217	0	<b>0.261</b>	<b>0.261</b>	1	$3 \rightarrow 1$
4	0.083	0.113	0.037	0	<b>0.767</b>	5	$4 \rightarrow 5$
5	0.085	0.121	0.036	<b>0.758</b>	0	4	$5 \rightarrow 4$

\*0 karena sudah melewati node tersebut



Node Awal	Probabilitas				
	1	2	3	4	5
1	0	0.28	0.129	0.29	0
2	0.23	0	0.088	0.327	0
3	0	0.217	0	0.261	0.261
4	0.083	0.113	0.037	0	0
5	0.085	0.121	0.036	0	0

Dengan nilai  $\tau_{ij}$  masih sama = 0.021 dan hasil rute sebelumnya diperoleh rute berikutnya

Node Awal	Probabilitas					Node Tujuan	Memori
	1	2	3	4	5		
1	0	0.4	0.185	<b>0.415</b>	0	4	$1 \rightarrow 5 \rightarrow 4$
2	0.356	0	0.137	<b>0.507</b>	0	4	$2 \rightarrow 5 \rightarrow 4$
3	0	0.294	0	0.353	<b>0.353</b>	5	$3 \rightarrow 1 \rightarrow 5$
4	0.355	<b>0.487</b>	0.158	0	0	2	$4 \rightarrow 5 \rightarrow 2$
5	0.35	<b>0.5</b>	0.15	0	0	2	$5 \rightarrow 4 \rightarrow 2$



$$p_{52}^5(0) = \frac{\tau_{52}(0) * \eta_{52}(0)}{\tau_{51}(0)*\eta_{51}(0) + \tau_{52}(0)*\eta_{52}(0) + \tau_{53}(0)*\eta_{53}(0) + \tau_{54}(0)*\eta_{54}(0) + \tau_{55}(0)*\eta_{55}(0)}$$

$$p_{52}^5(0) = \frac{0.121 * 0.021}{0.085*0.021 + 0.121*0.021 + 0.036*0.021 + 0*0.021 + 0*0.021} = 0.5$$

Hasil rute akhir pada iterasi 0:

Node Awal	Probabilitas					Node Tujuan	Memori
	1	2	3	4	5		
1	0	0	1	0	0	3	$1 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 3$
2	0	0	1	0	0	3	$2 \rightarrow 5 \rightarrow 4 \rightarrow 1 \rightarrow 3$
3	0	1	0	0	0	2	$3 \rightarrow 1 \rightarrow 5 \rightarrow 4 \rightarrow 2$
4	0	0	1	0	0	3	$4 \rightarrow 5 \rightarrow 2 \rightarrow 1 \rightarrow 3$
5	0	0	1	0	0	3	$5 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 3$

Hitung Cost Jarak dari hasil rute akhir:

$$f(x^1(0)) = d_{15} + d_{54} + d_{42} + d_{23} = 35.1 + 4 + 27 + 101 = 247.7$$

$$f(x^2(0)) = d_{25} + d_{54} + d_{41} + d_{13} = 24.9 + 4 + 36.9 + 80.6 = 247.4$$

$$f(x^3(0)) = d_{31} + d_{15} + d_{54} + d_{42} = 80.6 + 35.1 + 4 + 27 = 247.7$$

$$f(x^4(0)) = d_{45} + d_{52} + d_{21} + d_{13} = 4 + 24.9 + 39.1 + 80.6 = 233.5$$

$$f(x^5(0)) = d_{54} + d_{42} + d_{21} + d_{13} = 4 + 27 + 39.1 + 80.6 = 235.9$$

Hitung Perubahan Intensitas Pheromone tiap semut:

$$\Delta\tau_{ij}^k = Q / f(x^k(t)) \rightarrow \Delta\tau_{ij}^1 = 1 / f(x^1(0)) = 1 / 247.7 = 0.004037$$

$$\Delta\tau_{ij}^2 = 1 / f(x^2(0)) = 1 / 247.4 = 0.004042$$

$$\Delta\tau_{ij}^3 = 1 / f(x^3(0)) = 1 / 247.7 = 0.004037$$

$$\Delta\tau_{ij}^4 = 1 / f(x^4(0)) = 1 / 233.5 = 0.004283$$

$$\Delta\tau_{ij}^5 = 1 / f(x^5(0)) = 1 / 235.9 = 0.004239$$

### Matriks Perubahan Intensitas Pheromone semut 1 dan 4:

$\Delta\tau_{ij}^1$	1	2	3	4	5
1	0	0	0.004037	0	0.004037
2	0	0	0.004037	0.004037	0
3	0.004037	0.004037	0	0	0
4	0		0	0	0.004037
5	0.004037	0	0	0.004037	0

$\Delta\tau_{ij}^4$	1	2	3	4	5
1	0	0.004283	0.004283	0	0
2	0.004283	0	0	0	0.004283
3	0.004283	0	0	0.004283	0
4	0	0	0.004283	0	0.004283
5	0	0.004283	0	0.004283	0

Hitung Update Intensitas Pheromone (Ambil contoh i = 1, j = 2 dan i = 5, j = 2):

$$\Delta\tau_{ij} = \sum \Delta\tau_{ij}^k \rightarrow \Delta\tau_{12} = \Delta\tau_{12}^1 + \Delta\tau_{12}^2 + \Delta\tau_{12}^3 + \Delta\tau_{12}^4 + \Delta\tau_{12}^5 = 0+0+0+0.004283+0.004239 \\ = 0.008522 \quad (\text{Jumlahkan yang ada lewat di rute pada semut-k})$$

$$\rightarrow \tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij} \rightarrow \tau_{12}(t+1) = (1-0.9) * \tau_{12}(t) + \Delta\tau_{12} = 0.1 * 0.021 + 0.008522 \\ = 0.0021 + 0.008522 = 0.010622$$

$$\Delta\tau_{ij} = \sum \Delta\tau_{ij}^k \rightarrow \Delta\tau_{52} = \Delta\tau_{52}^1 + \Delta\tau_{52}^2 + \Delta\tau_{52}^3 + \Delta\tau_{52}^4 + \Delta\tau_{52}^5 = 0+0.004042+0+0.004283+0 \\ = 0.008325 \quad (\text{Jumlahkan yang ada lewat di rute pada semut-k})$$

$$\rightarrow \tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \Delta\tau_{ij} \rightarrow \tau_{52}(t+1) = (1-0.9) * \tau_{52}(t) + \Delta\tau_{52} = 0.1 * 0.021 + 0.008325 \\ = 0.0021 + 0.008325 = 0.010425$$

Hasil Update Semua Intensitas Pheromone:

$\tau_{ij}(t+1)$	1	2	3	4	5
1	0	0.010622	0.022738	0.006142	0.010174
2	0.010622	0	0.014216	0.014413	0.010425
3	0.022738	0.014216	0	0.006383	0.006339
4	0.006142	0.014413	0.006383	0	0.022738
5	0.010174	0.010425	0.006339	0.022738	0

∴ Diulang seperti contoh awal hingga iterasi max & nilai akhir yang diambil adalah cost jarak paling sedikit / nilai pheromone tertinggi.



# TAMBAHAN

Nodes	A	B	C	D	E	F	G
A	0	4.9	5.3	4.8	5.5	3.7	1.6
B	4.9	0	3.5	1.2	1.4	1.6	6
C	5.3	3.5	0	2.6	4.5	6.5	6.5
D	4.8	1.2	2.6	0	2.7	6	6
E	5.5	1.4	0	2.7	0	6.6	4.7
F	3.7	1.6	6.5	6	6.6	0	2.3
G	1.6	6	6.5	6	4.7	2.3	0

Jika tidak diberitahu nilai pheromon awal, maka dapat dilakukan dengan menggunakan Greedy Algorithm dengan pemisalan titik awal A.

Kemudian cari jarak A terpendek yaitu A ke G (1.6), lalu dari G terpendek ke F (2.3) (Tidak ke A karena akan ada loop), dan seterusnya.

Jumlahkan semua jalur terpendek diperoleh:

$$A \rightarrow G \rightarrow F \rightarrow B \rightarrow D \rightarrow C \rightarrow E \rightarrow A$$

$$= 1.6 + 2.3 + 1.6 + 1.2 + 2.6 + 4.5 + 5.5$$

$$= 19.3$$

$$\therefore \tau_{ij} = \tau_0 = 7 / 19.3 = 0.363$$

\*7 dari banyak node



## Links:

- <https://chatgpt.com/share/66f305ea-7bfc-8008-90ac-2e1da32680b7>
- <https://socs.binus.ac.id/2019/12/31/ant-colony-optimization/>
- <https://www.geeksforgeeks.org/introduction-to-ant-colony-optimization/>

## Paper:

- <https://jurnal.ugm.ac.id/ijccs/article/view/3052/2709>
- <http://e-journal.uajy.ac.id/11088/4/3MTF02407.pdf>
- <https://media.neliti.com/media/publications/93603-ID-none.pdf>
- <https://www.sciencedirect.com/science/article/abs/pii/S0142061515005840>
- [https://www.researchgate.net/figure/Flowchart-for-the-ACO-algorithm\\_fig1\\_308911799](https://www.researchgate.net/figure/Flowchart-for-the-ACO-algorithm_fig1_308911799)
- [https://www.researchgate.net/figure/Pseudo-code-for-the-Ant-Colony-System-ACS-algorithm\\_fig1\\_28600558](https://www.researchgate.net/figure/Pseudo-code-for-the-Ant-Colony-System-ACS-algorithm_fig1_28600558)
- [https://www.researchgate.net/figure/Pseudocode-of-Ant-Colony-Optimization\\_tbl2\\_347256298](https://www.researchgate.net/figure/Pseudocode-of-Ant-Colony-Optimization_tbl2_347256298)

## Video:

- <https://youtu.be/qfeymoF8pb4?si=GAZggint8pyLy5Q6>
- <https://youtu.be/783ZtAF4j5g?si=6C0QzsEZHh1PPHPt>